Microprocessors

- Microprocessors on the market
  - General-Purpose Microprocessor
    - Example: Intel Pentium 4
    - Main application: Personal computers
  - Microcontroller
    - Example: Microchip PIC16F87x
    - Main application: Embedded systems
  - Digital Signal Processor (DSP)
    - Example: Texas Instrument TMS320C6000
    - Main application: Digital signal processing systems

A Simple Microprocessor System

- Micro Processor Unit (MPU) or Central Processing Unit (CPU)
- Memory Unit
- Input/Output Unit (I/O Unit)
- Bus system
- Peripherals (Input/Output Devices)

Program, Data and Memory

- A computer works based on the idea that programs and data can be accessed (fetched and stored) in the memory.
- A program consists of a sequence of instructions.
- Each instruction is coded in a binary number (e.g., 8 bits, 16 bits or 32 bits).
- An instruction consists of an operation and some address fields.

Program, Data and Memory

- A data byte is also coded in an 8-bit binary number.
- A data word is also coded in a 16-bit binary number.
- One cannot distinguish a 16-bit instruction from a 16-bit data word when they are stored in the memory.

Central Processing Unit

- Execution Unit
  - Arithmetic and Logic Unit (ALU)
  - General-purpose registers
  - Status register (SR)
  - Program Counter (PC)
- Control Unit
  - Instruction register (IR)
  - Instruction decoder
- Internal Buses

by Qin-Zhong Ye
Program Execution

- The Program Counter (PC) holds the address of the next instruction to be executed.
- This address is used to fetch the next instruction from the memory to the Instruction Register.
- The instruction is decoded by the instruction decoder and a sequence of control signals is generated.
- The control signals enable various registers (i.e., fetch data) and ALU to perform an arithmetic operation or a memory data access operation.

Program Execution (continued)

- The result of the operation is stored in a register or a memory location.
- The program counter is incremented (or changed if the instruction is a jump instruction).

Computer Architecture

- von Neumann Architecture
- Harvard Architecture
- Pipeline
- Cache Memory
- CISC and RISC

Harvard Architecture

Instruction fetch and memory data access can be performed in parallel.
Pipeline

- Instruction pipeline

Instruction fetch → Instruction decode and register fetch → Execution and address calculation → Memory access → Write back

IF ID EX MEM WB

Cache Memory

- Memory Hierarchy
  - Registers, Cache memory, Main memory, Disk
- Cache memory is implemented by SRAM (Static Random Access Memory) which is faster than DRAM (Dynamic Random Access Memory).
- DRAM is used as the main memory because it is cheaper.

Cache Memory (continued)

- Cache memory is used to save frequently used instructions and data so that memory accessing time is reduced.
- The number of accesses to the main memory through the bus is reduced and therefore other devices can use the bus without affecting the performance of the CPU.

CISC and RISC

- CISC (Complex Instruction Set Computer)
- RISC (Reduced Instruction Set Computer)
- Computer History:
  - RISC → CISC → RISC
- Comparison of CISC and RISC
  - CISC has more instructions for programmers.
  - A single CISC instruction can perform more operations.
  - CISC instructions can be implemented by microprograms.
CISC and RISC
(continued)

- RISC has less instructions which are easy for programmers to learn and remember.
- RISC Instructions only perform simple operations.
- It is easier to pipeline the operations of the RISC instructions.
- Cache memory provides a good solution for fast memory accesses required by RISC.

CISC and RISC
(continued)

- Modern CISC processors use both the pipeline technique and cache memory.
- Modern RISC processors increase the number of instructions if the newly introduced instructions do not degrade performance.
- Dynamic execution is used to keep the order of results of the instruction executions as the original order of instructions in the program.